南京大学

本 科 毕 业 论 文

院　　系　计算机科学与技术系

专　　业　计算机科学与技术

题　　目　基于 BERT 预训练模型的
文本分类任务探究

年　　级　2018　学 号　181840070

学生姓名　葛睿芃

指导教师　张建兵　职 称　副教授
宗石　职 称　助理研究员

提交日期　2022 年 6 月 6 日

南京大学本科毕业论文（设计）
诚信承诺书

本人郑重承诺：本人葛睿芃所呈交的毕业论文（设计）（题目：基于 BERT 预训练模型的文本分类任务探究）是在指导教师张建兵、宗石的指导下严格按照学校和院系有关规定由本人独立完成的。本毕业论文（设计）中引用他人观点及参考资源的内容均已标注引用，如出现侵犯他人知识产权的行为，由本人承担相应法律责任。本人承诺不存在抄袭、伪造、篡改、代写、买卖毕业论文（设计）等违纪行为。

作者签名：葛睿芃

学号：181840070

日期：2022-06-06

南京大学本科生毕业论文（设计、作品）中文摘要

题目：基于 BERT 预训练模型的文本分类任务探究

院系：计算机科学与技术系

专业：计算机科学与技术

本科生姓名：葛睿芃

指导教师（姓名、职称）：张建兵 副教授　　宗石 助理研究员

摘要：

大数据时代下，新闻、评论类短文本数据在网络上日益增多。对于短文本数据，实现基于话题、情感的自动分类，可以帮助用户识别有价值的信息。传统的文本分类方法在特征提取和分类模型的搭建上存在一些问题，例如文本特征过于稀疏，不能表示单词间的语义关系等。BERT 预训练模型提供了一种基于端到端的分类方式，内部由 12 层双向 Transformer 结构组成。BERT 在大规模语料中预训练后并进行微调，便可以达到较好的分类效果。

本文对 BERT 预训练模型在短文本分类这一下游任务的细节进行研究。应用 BERT 完成文本分类任务一般需要经过文本预处理、增强预训练和模型微调三个阶段。在一般的增强预训练算法基础上，我们针对类别不平衡等问题对算法提出改进。本文比较 BERT 模型与经典模型在文本分类任务上的结果，验证了 BERT 应用于文本分类的突出表现。其次，本文还对 BERT 模型各层的输出表示进行观察分析。通过实验，我们验证了较低层次的输出更加关注语法等表层特征，而较高层次的输出更加关注语义等深层特征；在网络自主学习各层权重的情况下，高层次的输出会获得更高的权重。最后，本文对数据集样本量较低的情况下的分类结果进行了研究。BERT 可以在一定程度上削减样本数据量低带来的不利影响，且增强预训练阶段会进一步提高分类表现。通过对 BERT 的预训练机制进行分析，以及对训练过程进行观察，我们验证了这一现象在短文本分类中也是存在的。

关键词：短文本；分类；预训练；BERT

南京大学本科生毕业论文（设计、作品）英文摘要

THESIS: Research in Short Text Classification Based on Pre-trained Model BERT

DEPARTMENT: Department of Computer Science and Technology

SPECIALIZATION: Computer Science and Technology

UNDERGRADUATE: Ruipeng Ge

MENTOR: Associate Professor Jianbing Zhang　Research Assistant Shi Zong

ABSTRACT:

In the era of big data, short text such as news and comments are growing significantly through the Internet. It is thus important to design topics or sentiment classification models to automatically identify valuable information. Traditional text classification methods have problems regarding feature extraction and model structures, such as sparsity of features, or lack of semantic relations. Pre-trained BERT model provides an end-to-end paradigm. It consists of 12 bi-directional Transformer structures, and can achieve high classification performance after being pre-trained in large corpus and fine-tuned.

This paper study short text classification with pre-trained model BERT. There are usually three phases regarding text classification with BERT, including Pre-processing, Further Pre-training and Fine-tuning. We make an improvement to the algorithm of Further Pre-training, especially addressing the problem of class imbalance. We compare classification performances of BERT and other baseline models and proved the exceeding performances of BERT model. Moreover, we observe and analyze the output of each layer of BERT model. By conducting experiments, We verify that outputs of lower layers focus more on low-level features like syntax, while those of higher layers focus more on high-level features like semantics. Weights of higher layer outputs tend to be greater when we make those weights learnable. Finally, we focus on the condition where the number of samples in training set is low. BERT can weaken the disadvantages of low quantity of data to a certain extent, and the Further Pre-training phase will contribute to higher performances. We analyze the pre-training mechanism as well as the training process, and verify the existence of such phenomenon.

Keywords: Short Text; Classification; Pre-training; BERT

目　录

第一章　绪论

1.1　研究背景及意义

在互联网技术飞速发展的今天，网络上的数据资源的数量正在以惊人的速度增长，并且成为人们获取信息的主要手段。进入大数据时代，数据资源更是呈现出大体量、高速度和高多样性的特征[1]。如今的互联网已经成为人们日常生活不可分割的一部分，人们逐渐开始放弃原来从纸质媒介获取消息的方式，而转为更加快捷方便的电子数据资源。从数据挖掘的角度，大量的数据资源中蕴含着许多有价值的信息，而如何使计算机能够自动从大数据中识别未知的、有效的、有行为价值的信息，并以此辅助人们做出重要决策，是当前研究的热点问题之一[2]。用户在互联网中的行为都会产生相应的数据，新闻、电商、短视频、直播网站可以通过分析用户行为，自动做出信息或商品的精准推送，让用户方便地看到自己感兴趣的信息，从而引导用户消费，提高企业的利润[3]。

自然语言文本是网络上最常见的数据资源之一，从内容上看包括新闻文本，聊天文本，用户点评等。对于新闻文本而言，这些新闻需要通过分类展示的方式显示在新闻网站上，供不同用户根据自己的要求和兴趣筛选阅读。但是，庞大的新闻数量使得人工分类变得极为困难，新闻网站将更多地依赖计算机自动进行新闻分类。对于评论而言，通过计算机自动识别评论的情感倾向，更加有利于用户对于评论主题质量的识别。计算机通过训练人工智能模型，可以实现文本匹配最相关的话题或者情感标签，并且随着计算机硬件性能和模型复杂度的提升，匹配的准确率正在逐步提升。文本分类的自动化能够帮助新闻网站管理者更好地维护网站内容，节约大量人力，同时能够方便用户查找对自己有用的消息，提升用户的使用体验。

新闻、评论这类短文本作为独特的自然语言文本，从数据挖掘的角度上属于一种复杂类型的数据。机器学习领域已经广泛应用到文本数据挖掘中，但是受制于文本数据高维、稀疏、格式不确定的特点，机器学习模型的效果受到了

比较大的制约[4]。传统的基于特征选取的文本分类方法缺乏对词语的语义信息的表示，也很难结合长距离上下文的语义信息作出分类分类判断，这种分类方式制约了分类模型的性能。为了解决这一问题，依靠深度学习模型的“端到端”分类方式成为了文本分类领域的热点问题。

从现实的角度，新闻、评论等短文本往往具有很大的时效性和准确性，这要求分类模型同时具备较高的准确度和运行效率。模型性能的提高将在一定程度上提升用户的体验。同时，随着预训练模型在自然语言处理领域的广泛应用，不同的下游任务可以通过针对特定数据集，在预训练模型微调的方式达到较好的性能，实现模型在不同任务之间的迁移[5]。在文本分类任务的基础上，还可以将模型迁移至其他文本挖掘任务，如自动聚类，异常值自动监测等，并可应用到诸如舆情监测、垃圾邮件识别等领域，表现出与传统方法更强大的性能。

1.2 研究现状

一般而言，文本分类任务需要经过以下步骤：文本预处理、提取文本特征、建立文本模型、模型训练和测试。在传统方法中，文本分类通常包括两个关键点：选取合适的特征，以及建立合适的模型。

1.2.1 文本特征选取

分类模型需要接受向量化的文本特征作为输入，因此，需要寻找合适的指标构建文本向量。Mikolov 等人[6]所提出的词袋模型（Bag-of-words, BOW）给出了一种简单的处理方式，他用长度为 N 的向量来表示一段文本，每一个维度代表词典里的一个词语，该维度下的数值代表这个词语在该段落中的出现次数。这种方式处理简单，但是存在一定的问题。首先，这种表示维度太高，且过于稀疏，不利于模型处理；其次，这种表示下字典中的每一个词语地位对等，无法处理一义多词、一词多义的情形，无法体现文本之间的语义关系。

针对维度过高的问题，可以通过特征提取（feature extraction）的方法进行降维。Shah 等人[7]总结了自然语言处理领域常见的文本提取方法。主成分分析（Principle Component Analysis，PCA）[8]是一种常见的降维方式，其原理是通过正交变换提取特征维度之间的相关性，并重新组合出一组低维无关的特征，达到

降维的效果。潜在语义分析（Latent Semantic Analysis，LSA）[9]是另一种常见的降维方式，它通过对词语-文本矩阵作奇异值分解，将文本向量映射到维度更小的潜在语义空间中，以达到降维的效果。

针对文本语义关系，Mikolov 等[10]提出的 Word2Vec 词嵌入模型通过训练神经网络来获得隐层表示，作为词向量。Word2Vec 模型有两种训练方式，CBOW 方式通过周围词预测目标词，而 Skip-gram 方式反过来通过目标词预测周围词，这样训练出来的词向量对于上下文的语义信息有着较好地把握。

1.2.2 文本分类模型

早在 20 世纪 90 年代，传统机器学习模型就已经运用在文本分类上，并取得了不错的效果[11]。朴素贝叶斯分类器[12]是一种较为简单的分类模型，它以特征维度间互相独立为主要假设，通过概率方法学习输入到输出的概率分布，基于此分布做出预测。支持向量机（Support Vector Machine，SVM）[13]是另一种常用的分类器，它通过最大化间隔的方式，生成用于分类的超平面，并可以通过核函数的方法实现非线性分类。

深度学习时代，更加复杂、深层的文本分类模型提高了分类的准确度和性能。卷积神经网络（Convoluted Neural Network，CNN）[14]通过对词组组成的特征矩阵进行卷积操作，构建分类模型；循环神经网络（Recurrent Neural Network，RNN）[15]则善于处理长序列信息，但由于训练时容易产生梯度爆炸或消失的问题，长短期记忆网络[16]（Long Short-Term Memory，LSTM）通过门控制的方式，对长距离上下文的信息进行选择性的记忆和遗忘，不仅解决了梯度爆炸和消失的问题，也能更好地利用上下文的信息。

随着硬件水平的提升，计算机训练深度模型的能力也得到大幅提高，随即出现了预训练模型。Devlin 等人[17]提出的 BERT 预训练模型是其中最为著名的，在多项自然语言处理任务中表现出色。它的基本组件为双向 Transformer 模型，Vaswani 等人[18]提出的 Transformer 是一种利用了多头注意力机制的语言模型，在文本语义特征的提取中有着较强的能力。BERT 预训练模型堆叠了多层 Transformer 结构，并利用大规模文本数据在其中建立预训练任务，此外，根据不同的下游任务，对模型展开进一步的微调（fine-tuning），以此来适应不同的文本挖掘任务。针对文本分类任务，Sun 等人[5]提出了一种通用的 BERT 模型微调

策略，即强化预训练、多任务学习、目标任务微调三步走的方式，并取得了较好的分类效果。Yang[19]对 Transformer 模型在文本分类中的应用进行了考察，并发现 Transformer 预训练模型在提取长距离上下文信息的能力上不如 RNN 结构的模型强，但是在 RNN 结构模型中嵌入多头注意力机制后，模型的能力得到了显著的提升，由此验证了注意力机制在提升分类模型性能中的有效性。

1.3 研究内容及方法

本文研究的主要内容就是探究新闻、评论等短文本数据的分类任务解决方法。在使用传统机器学习模型进行分类的时候，受制于文本表示的高维、稀疏特征，模型训练过程中一般会选择特征降维的方式，但因为这种方式往往会出现复杂度过高、特征丢失以及过拟合等现象，分类效果往往不尽如人意。结合目前较为先进的 BERT 预训练模型，使用 BERT 模型提取文本特征后，利用深度学习网络进行分类，将有效解决这一问题，提高模型的分类准确率和泛化能力。

本文将主要就 BERT 预训练模型在处理“文本分类”这一下游任务时的具体方法与分类表现进行探究。我们通过研究经典文本特征表示和文本分类模型，分析其中的不足之处，引出 BERT 预训练模型的优势，根据相应的模型训练方法，作出相应的改进，并通过在多个文本数据集上进行测试，证明该模型的优势。具体而言，本文将：

（1）研究 BERT 模型的基本结构和原理，分析其优点。同时，从文本预处理、长文本截断、增强预训练、微调四个角度，探究 BERT 模型应用于文本分类的基本方法。

（2）在 4 种英文、中文文本数据集上，应用 BERT 预训练模型和多种下游分类模型进行训练，并与传统深度学习模型网络分类表现进行比较，验证应用 BERT 预训练模型的分类模型在分类性能上较传统模型有明显提升。

（3）针对 BERT 模型的内部结构特点，以“BERT 模型的各层输出表示”和“低样本量数据情况”为关注点，对 BERT 模型进行深入分析和实验验证，总结应用 BERT 预训练模型进行文本分类的一般方法。

1.4 主要章节安排

本文的主要章节安排如下所示：

第一章为绪论，介绍短文本自动分类这一需求产生的社会背景，以及研究短文本分类的意义；同时介绍文本特征选取与分类模型的研究现状，并对本文的研究内容，和各章安排进行简要介绍。

第二章将就文本分类任务及其方法进行介绍，首先介绍文本分类任务的定义和评价指标，然后介绍常见应用于文本分类的特征表示和深度学习模型，最后介绍 BERT 预训练模型的相关内容，包括其内部双向 Transformer 结构和两个预训练任务。

第三章主要介绍了 BERT 预训练模型如何应用短文本分类任务，将主要从短文本的预处理、长文本截断、模型预训练、微调训练四个方面进行介绍。短文本的预处理主要将介绍如何将短文本“符号化”并进行索引；长文本截断介绍如何截断文本以满足 BERT 的输入限制；模型预训练部分将主要介绍一种改进的“增强预训练”方式，用于提高分类效果；微调训练将介绍 BERT 预训练模型处理“文本分类”这一下游任务时的训练方法。

第四章将介绍实验细节、实验过程和实验结果。首先将介绍实验所使用的编程和运行环境，所使用数据集的具体情况，以及模型、训练过程的具体细节，并展示实验结果，同时与参考文献的基准模型方法进行对比，体现 BERT 预训练模型在分类表现上的优势。最后讨论实验结果，重点对 BERT 模型不同层次的输出，以及低样本量的情况展开讨论。

第五章将对本文内容进行总结，并根据实验不足进行展望。

第二章　文本分类方法

本章将介绍“文本分类”任务的定义和一般的解决方法。首先将对文本分类任务作简要概述，从数学语言的角度介绍文本分类的定义，以及评价文本分类模型优劣的方法。随后，从文本特征表示和分类模型两个方面，介绍传统机器学习方法和深度学习模型是如何解决这一问题的。最后，介绍 BERT 预训练模型的基本结构和训练方法。

2.1　文本分类任务概述

2.1.1　文本分类任务过程

分类任务是机器学习领域的一个重要内容，它的核心在于从训练样本中学习出一种分类模型，随后便可以利用这一个分类模型，对测试样本中的数据预测其标签[20]。对于短文本这一类文本数据而言，根据最终具体任务的不同，这个标签可以为内容话题、情感态度等不同类别。若要应用传统机器学习模型来解决文本分类任务，首先需要明确文本数据集中的“特征”。也就是说，需要将自然语言文本转化为特定的数学表示（一般为向量表示），这个过程就是文本特征化。而建立文本特征到标签的映射，就是训练分类模型的过程。现在，深度学习已经成为普遍应用的方法，它通过搭建深层次的复杂神经网络来训练复杂模型，并且得益于计算水平的提升和大数据的支撑，深度学习的训练已经变得十分高效，分类表现也有了显著提高[20]。

从数学语言的角度，文本可以看作是一系列单词（或者字、词组）的序列。对于一个文本数据 $\boldsymbol{D} = \{w_1, w_2, ..., w_N\}$，其可以表示为一个序列的形式。其中 $w_n (1 \leq n \leq N)$ 表示一个单词，N 表示文本的总长度。文本特征化就是通过某种映射方式，将文本映射为特征矩阵 $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$，作为模型的输入。其中 $\boldsymbol{x_n} \in \mathbb{R}^d (1 \leq n \leq N)$ 为单词 w_n 对应的向量表示，d 为词向量的维度。建立并训练分类模型后，通过将文本词向量输入到分类模型，即可预测文本的类别，

该过程可以表示为：

$$Y = f(\boldsymbol{X}) \mapsto \{0, 1\}\text{。} \tag{2.1}$$

其中 Y 为预测出的文本 D 的类别，f 为训练出来的分类模型映射。

在一般的文本分类任务中，单个文本的标签通常只包含一个类别，这种分类任务称为“单标签分类”。若最终单个文本的标签包含多个类别，则成为“多标签分类”，本文只讨论“单标签分类”的情况。

2.1.2 文本分类评价指标

为了定量表示模型的分类效果，需要通过量化评价指标的方式评价模型泛化性能。给定包含 M 个样本的测试数据集 $S = \{(D_1, Y_1), (D_2, Y_2), ..., (D_M, Y_M)\}$，其中 Y_m 为文本 $D_m (1 \leq m \leq M)$ 的真实类别，模型 f 在测试数据 D_m 上的预测标签为 $f(\boldsymbol{X_m})$，其中 $\boldsymbol{X_m}$ 是文本 D_m 的特征表示。评价指标需要衡量预测标签与真实标签之间的匹配程度，以此来衡量模型的泛化能力。

常见用于分类任务的模型评价指标有分类精度和 F1 分数[20]。分类精度是指分类正确的样本（即预测标签和真实标签完全匹配的样本）占总样本的比例，即：

$$\text{acc}(f; S) = \frac{1}{M} \sum_{m=1}^{M} \mathbb{I}(f(\boldsymbol{X_m}) = Y_m), \tag{2.2}$$

与之相对的是分类错误率，即预测标签和真实标签不匹配的样本占总样本的比例，错误率越低，分类效果越好。

F1 分数则更能体现出分类模型对于查全和查准能力的综合表现。对于二分类任务，标签可以分为正例和反例两个类别。定义查准率（或准确率）P 为所有预测结果为正例的样本中真正例的比例，查全率（或召回率）R 为所有真实标签为正例的样本中预测也为正例的比例，则模型的 F1 分数可以表示为查准率 P 和查全率 R 的调和平均数，即：

$$\text{F1}(f; S) = \frac{2PR}{P + R}, \tag{2.3}$$

对于多分类任务，则可以分别取标签的每个类别作为正例，其他类别均作为反例，分别计算模型在这个类别中的查准率 P 和查全率 R，取所有类别的查准率、

查全率的平均值作为宏观查准率和查全率，以此计算出 F1 分数。这种方式计算出的分数被称为“宏 F1 分数”。在论文的实验部分，我们将采用“宏 F1 分数”作为评价指标，衡量各个模型的泛化能力。

2.2 常见文本特征表示

计算机不能直接处理文本，因此我们需要将文本转化为特定的编码表示。在其他领域的深度学习任务中，特征维度往往比较明显，例如对图像的处理中，图像各像素点的颜色、灰度等都可以作为图像的特征表示，但是文本数据的编码不是非常直接。因此，需要设计出一种能够用于文本的特征表示。

以下几种都是较为常用的文本特征表示方法，在本节中，我们将具体介绍这些特征表示，并分析它们各自的优缺点。

2.2.1 基于统计方法的文本特征表示

文本可以看作是一系列单词的集合，在包含 N 个文本的文本集合上，对单词应用统计方法，就可以用特征向量的方式来表示文本。常见基于统计方法的文本特征表示有词袋模型（Bag-of-words, BOW）和词频-逆文本频率指数（Term Frequency-Inverse Document Frequency, TF-IDF）。

BOW 模型中，一个文本集合中出现的所有词汇将构成一个词典，文本的特征维度与词典中词的个数一致，每一个维度对应一个单词。BOW 模型通过统计文本中各个单词出现的次数来建立文本的特征表示，即每个维度的数值为对应单词在文本中的出现次数[21]。

TF-IDF 特征维度和 BOW 模型相同，其在考虑词频的同时，也考虑了单词在文本集合整体的出现频率。用公式表示即为：

$$\mathrm{TF}(w, D) = \frac{\mathrm{count}(w, D)}{\mathrm{size}(D)}, \tag{2.4}$$

$$\mathrm{IDF}(w) = \log \frac{N}{\mathrm{contain}(w)}, \tag{2.5}$$

$$\text{TF} - \text{IDF}(w, D) = \text{TF}(w, D) \times \text{IDF}(w), \tag{2.6}$$

其中，词频指数 TF 记录单词 w 在文本 D 中出现次数与文本总单词数的比值，逆文本频率指数 IDF 记录文本集合中所有出现该词的文本的比例的倒数。单词在文本中出现的频率越高，在其他文本中出现次数越少，则 TF-IDF 系数越高。TF-IDF 特征表示的每一个维度的数值即为该单词对应的 TF-IDF 数值。

两种基于统计的特征表示都十分简单、灵活，但是存在一定的缺陷。一是特征过于稀疏：在词典单词量很大的情况下，特征向量的维度也会因此增加，而一个文本所包含的单词数量要远低于这个数值，因此会造成文本特征过于稀疏，模型计算量增大的负面影响；二是没有体现文本的词序和语义特征：在 BOW 和 TF-IDF 特征表示中，所有文本均被看作成了单词的“集合”，而非“序列”，词的位置信息在特征中完全没有体现出来，同时，各个单词之间的地位是对等的，这种特征表示也无法处理单词之间的语义信息。

2.2.2 基于神经网络的分布式文本特征表示

深度学习时代，出现了基于神经网络的分布式文本特征表示。这种方式的核心思想是将每一个单词映射到较低维度的向量空间中，降低维度的同时，保留词之间的语义关系。这种特征表示可以避免输入特征维度过高、过于稀疏，且不至于丢失过多的语义信息。

Mikolov 等人[10]所提出的 Word2Vec 模型是一种最常见的分布式文本特征表示方法。Word2Vec 搭建神经网络，通过在特定任务上训练模型的方式，取模型的隐层表示来获取文本特征，以此将各个单词映射到向量空间中。Word2Vec 可以与深度学习模型灵活结合，方便地获取到文本的特征表示，在深度学习和自然语言处理领域具有非常广泛的应用。

Word2Vec 有两种模型：Continuous Bag-of-words(CBOW) 模型和 Skip-gram 模型。这两种模型如图 2.1 所示：

CBOW 模型的训练任务是根据目标位置上下文的单词，预测目标位置的单词。在实际操作中会引入一个长度为 $2c+1$ 的滑动窗口，根据中心位置上下各 c 个单词预测中心词（图 2.1 中取 $c=2$）。CBOW 网络包含一个输入层、一个投影

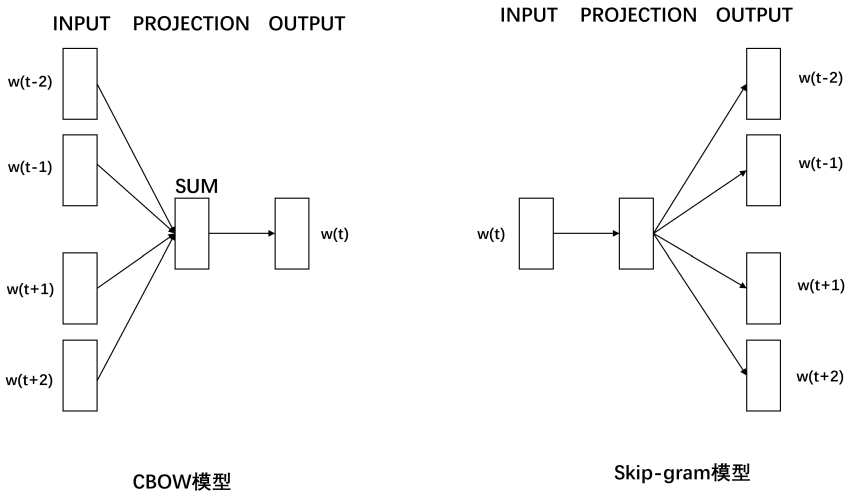

图 2.1 Word2Vec 模型

层和一个输出层，输入的单词以独热编码（one-hot）形式，经过一个线性层后得到各自的输出后，对各个滑动窗口中 $2c$ 个上下文单词的输出全部相加，再次经过一个线性层得到预测词的输出矩阵，通过激活函数 softmax 得到中心词的预测概率，从而选出最有可能的单词。CBOW 模型使用交叉熵（cross entropy）损失函数，通过反向传播（back propagation）更新网络的参数。在达到规定的迭代次数后，任意一个单词的独热编码，通过第一个线性层的权重矩阵作用后，就可以得到这个单词的词嵌入表示（word embedding），即特征表示。

skip-gram 模型的结构和 CBOW 十分类似，训练过程也基本相同，但是执行的是相反的训练任务——即通过中心词预测窗口中上下 c 个单词。skip-gram 的输入层是窗口内中心词的独热编码向量，输出的是上下 c 个单词的预测概率。与 CBOW 类似，skip-gram 在训练完成后，其第一个线性层的权重矩阵就可以用来获取任意单词的词嵌入表示。

Word2Vec 词嵌入特征表示有效地解决了传统方法词向量维度过高和稀疏的问题，同时考虑了上下文语义关系，网络结构简单，通用性较强，能够应用于自然语言处理领域中的很多任务。但同时，Word2Vec 模型训练出来的词向量是静态的、一对一的，无法处理一词多义的情形。

2.3 常见文本分类模型

在获得了文本的特征表示后，文本分类模型就可以通过深度提取文本的语义信息，并根据这些信息做出判断，完成分类。传统的分类方法使用一些简单数学模型和算法来完成分类过程，例如贝叶斯分类、支持向量机（SVM）、对数几率回归（Logistic Regression）等。在深度学习兴盛的如今，搭建深度神经网络来解决这一问题已经成为主流的方法。深度神经网络以强大的算力为代价，通过复杂、深层的网络结构，充分地学习文本的特征信息以辅助分类决策，具有很强的泛化特征。

在文本分类任务中，最为常用的深度学习模型有卷积神经网络（Convolutional Neural Network, CNN），循环神经网络（Recurrent Neural Network, RNN），和基于注意力（Attention）机制的 Transformer 结构。

2.3.1 卷积神经网络结构

CNN 网络最初在计算机视觉领域较为流行，擅长处理二维图像信息。在处理文本的时候，我们可以将文本中各个单词的特征向量按序拼接，形成一个二维矩阵。通过类似处理图像的方法，就可以对这个文本矩阵进行处理。利用 CNN 结构处理文本数据的模型被统称为 TextCNN。TextCNN 的基本结构如图 2.2 所示。

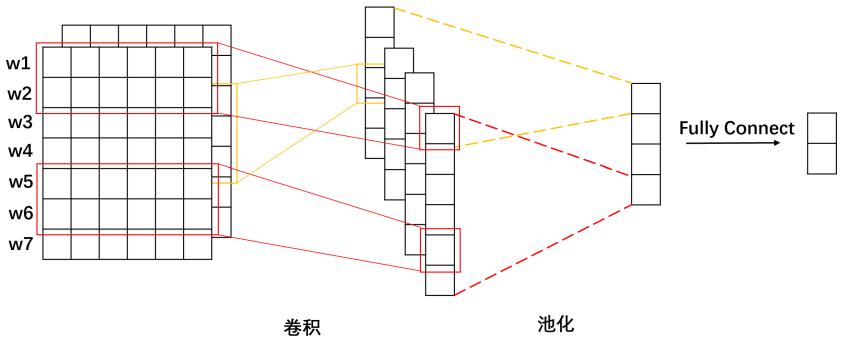

图 2.2 TextCNN 基本结构

CNN 结构的两个关键操作是卷积和池化。卷积操作需要一个固定大小的卷积核，从二维数据的左上角开始逐行扫描，每次被覆盖区域数据与卷积核上的权值进行相乘并求和，得到的值按照位置相互拼接，就获得了卷积后的特征矩

阵。在 TextCNN 中，卷积核在词向量方向上的维度通常选择与词特征向量维度一致，通过这种方式进行卷积操作后，其特征就变成了文本的 *n*-gram 特征表示。

池化的目的是降低特征维度，最常见的有最大池化（max pooling），即在特征矩阵的一个区域内仅保留最大值，去除其他特征，经过池化后的特征通过一个全连接层，经过激活函数，即可输出最终的分类结果。

CNN 在训练时往往采用并行多通道的策略，借鉴图像处理中的 RGB 三通道的方式，对多个通道的输入进行卷积操作，然后再进行合并，这种策略可以同时处理文本数据在多个方面的特征。另外，CNN 结构的模型在卷积层则采取“权值共享”的策略，降低了网络的参数大小。

CNN 结构的模型训练时高效，能达到一定泛化能力，但是由于不同输入的运算相互独立，CNN 结构训练是不会记忆之前时间的结果，因此和序列相关的特征在一定程度上会被丢弃。

2.3.2 循环神经网络结构

RNN 结构具有递归的特点，能够保留之前时间序列的信息，用于后续的训练，这种特点非常适合文本这种序列性的数据处理。RNN 可以接受任意长时间序列的信息作为输入，同时记录上一个时间节点的隐层表示，输出的结果将作为下一个时间节点的隐层输入。RNN 结构展开表示后，可以看作一系列抽象节点相互连接，但不同时刻共用一套参数。

用于处理文本任务的 RNN 结构被统称为 TextRNN 结构，TextRNN 结构具体如图 2.3 所示：其中，$\boldsymbol{X}$ 为输入的文本特征矩阵，$\boldsymbol{O}$ 为输出结果，$\boldsymbol{h}$ 为隐层表

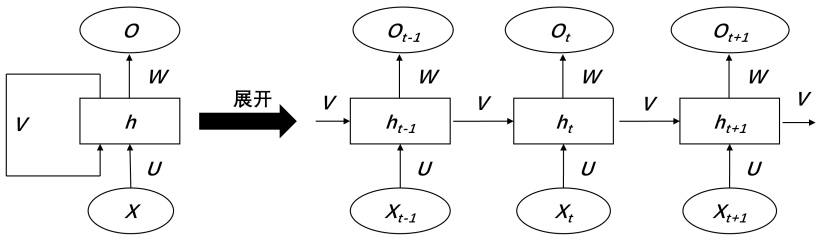

图 2.3 TextRNN 基本结构

示，这些都是和时间序列有关的；矩阵 $\boldsymbol{U}$，$\boldsymbol{V}$，$\boldsymbol{W}$ 为网络参数。

利用数学公式，RNN 结构前向传播的递推过程可以表示为：

$$\boldsymbol{h}_t = \phi(\boldsymbol{U}\boldsymbol{X}_t + \boldsymbol{V}\boldsymbol{h}_{t-1} + b), \tag{2.7}$$

$$\boldsymbol{O}_t = \boldsymbol{V}\boldsymbol{h}_t + c, \tag{2.8}$$

其中，ϕ 函数为激活函数，在 RNN 中通常选择 tanh 函数，b 和 c 为偏置量。

RNN 结构通常取最后一个时间的输出作为循环层的输出表示，当 TextRNN 应用于文本分类时，最后得出分类结果 $\boldsymbol{Y}$ 的过程可以表示为：

$$\boldsymbol{Y} = \text{softmax}(\boldsymbol{O}_t). \tag{2.9}$$

RNN 结构在使用反向传播更新参数时，回将模型展开，按照时间序列的顺序返回，按照一定的误差函数和梯度下降方法更新参数。在计算时，每一个时间步的误差将会全部相加，得到最终整个网络的误差。根据求导的链式法则，在梯度计算的过程会存在将多个时间步的梯度累乘的运算，此时如果任意一个时间步的梯度为 0，则整体梯度会迅速趋向于 0，产生“梯度消失”现象；如果各时间步的梯度都很大，此时计算出来的梯度会趋于无穷大，产生“梯度爆炸”现象。为了解决这些问题，Hochreiter 和 Schmidhuber[16]提出了长短时记忆网络（Long Short Term Memory, LSTM），通过门控制的方法解决这一问题。

LSTM 网络的具体结构如下图所示，其包含遗忘门、输入门、输出门 3 个门控结构，除了隐层状态 $\boldsymbol{h}$ 外，还拥有一个细胞状态参数 $\boldsymbol{c}$，其单元内部结构图如图 2.4 所示：

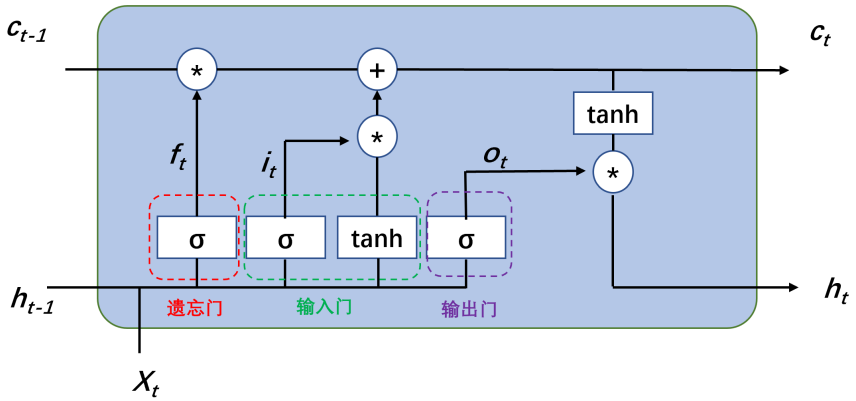

图 2.4 LSTM 单元基本结构

遗忘门用于确定上一个时间步的隐层表示有多少内容需要被舍弃，这部分读取上一个时间步的隐层 $\boldsymbol{h}_{t-1}$ 和当前时间步的输入 $\boldsymbol{X}_t$，输出部分将用于输入门，用数学公式表示即为：

$$f_t = \sigma(\boldsymbol{W}_f[\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + b_f), \tag{2.10}$$

其中 σ 为 sigmoid 激活函数，$\boldsymbol{W}_f$ 是遗忘门的参数，b_f 为遗忘门的偏置量，输出的结果 f_t 取值范围为 [0, 1]，用于决定细胞状态的信息保留程度。

输入门包括两个部分，第一部分用于决定输入数据的保留比例，第二部分用于更新细胞状态，数学公式表示为：

$$i_t = \sigma(\boldsymbol{W}_i[\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + b_i), \tag{2.11}$$

$$\hat{\boldsymbol{c}}_t = \tanh(\boldsymbol{W}_c[\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + b_c), \tag{2.12}$$

$$\boldsymbol{c}_t = f_t * \boldsymbol{c}_{t-1} + i_t * \hat{\boldsymbol{c}}_t, \tag{2.13}$$

其中，在更新细胞状态 $\boldsymbol{c}_t$ 时，接收了遗忘门的输出 f_t，决定 $\boldsymbol{c}_{t-1}$ 中信息保留的比例；输入门的另一个输出 i_t 则决定了输入数据经过本层神经元结构后 $\hat{\boldsymbol{c}}_t$ 中信息在细胞状态中的保留比例；两部分数据相加后就决定了这个时间步后的细胞状态 $\boldsymbol{c}_t$ 的值。

输出门将决定本时间步的隐层输出态 $\boldsymbol{h}_t$，隐层输出将取自本时间步的细胞状态态 $\boldsymbol{c}_t$，且只有一部分信息将会被保留，和输入们类似，这个权重也是由输入和上一时间步的隐层状态决定的，其数学公式如下所示：

$$o_t = \sigma(\boldsymbol{W}_o[\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + b_o), \tag{2.14}$$

$$\boldsymbol{h}_t = o_t * \tanh(\boldsymbol{c}_t), \tag{2.15}$$

其中，o_t 决定了细胞状态 $\boldsymbol{c}_t$ 中的信息保留比例，最终输出本时间步的隐层表示

$\boldsymbol{h}_t$。

LSTM 解决梯度问题的关键就是有效解决了长时依赖下，梯度趋近于 0 或者无穷大的情况；通过网络学习门控参数的方式，使得计算时的梯度不会一直处于小于 1 或大于 1 的状态，以此来调整梯度不至于消失或爆炸。同时，参数的增加也有利于网络对文本信息进行深层次的挖掘，这也让 LSTM 成为 RNN 结构中最常使用的模型结构之一。除了 LSTM 外，还有 GRU 结构也通过引入门控的方式来解决 RNN 梯度问题，但 GRU 使用的参数比 LSTM 少，结构较 LSTM 简单。

2.3.3 Transformer 结构

在自然语言处理领域，使用 CNN 或者 RNN 结构处理文本任务时，对于语义信息的学习有一定局限性。例如在 CNN 结构中，提取的 n-gram 特征仅在一个有限的上下文范围中学习其中的语义联系，但是对于更长距离的上下文信息却无法学习。为了解决自然语言文本中的语义联系问题，提升模型的泛化能力，我们常常使用注意力机制（Attention）来解决这个问题。

注意力机制可以用于任何序列数据的处理，其核心是让网络自动学习序列中对不同部分的关注程度。具体而言，首先应用模型将文本 $\boldsymbol{D}$ 中的每一个单词 $\boldsymbol{X}_k$，输出其对应的提取特征后的输出向量 $\boldsymbol{w}_k$，然后通过学习出来的注意力权重 a_k 来作用于所有向量，求出加权线性和作为整个文本数据的特征表示。注意力机制在序列到序列的模型中用得较多，在分类模型中也有一定的应用。

Vaswani 等人[18]引入的 Transformer 模型在自然语言处理领域获得了非常广泛的应用。Transformer 结构最初应用于序列到序列的模型中，由其是机器翻译领域。Transformer 结构由编码器（Encoder）和解码器（Decoder）两部分组成，其中 Encoder 部分的结构可以借鉴于文本分类中，用于提取文本的特征。

Transformer 结构的 Encoder 部分如图 2.5 所示。其中包含 2 个主要结构：多头自注意力（Multi-head Self-attention）结构、前馈（Feed Forward）结构。

Transformer 结构对于输入单词的编码做了一些优化，除单词编码外，引入“位置编码”（Positional Embedding）的概念，帮助 Transformer 结构认识文本中的语序关系。Transformer 模型中使用了三角函数的线性变换来表示这一位置信息。最终输入到编码器部分的向量，即为词向量和位置向量的和。

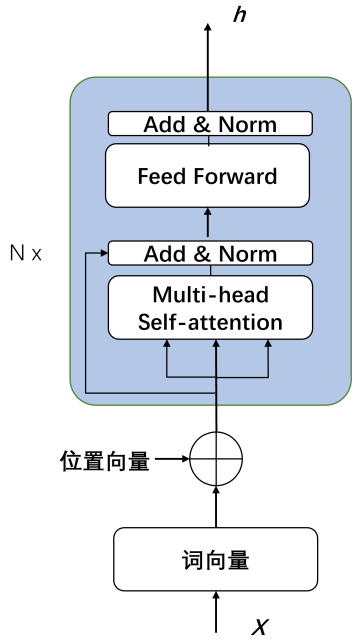

图 2.5 Transformer Encoder 基本结构

在多头自注意力结构中，网络将学习每个单词与全文其他单词的注意力权重，并以此来提取特征。一般的注意力机制需要三种输入向量：查询向量（Query）、键向量（Key）和值向量（Value），通过计算 Query 与 Key 之间的权重，并让 Value 以此权重线性求和，即可完成注意力机制的特征提取。在 Transformer 的自注意力机制中，Query、Key、Value 向量将分别由 3 个权重矩阵 $\boldsymbol{W_Q}$、$\boldsymbol{W_K}$ 和 $\boldsymbol{W_V}$ 作用于输入文本矩阵后得到，分别得到矩阵 $\boldsymbol{Q}$、$\boldsymbol{K}$ 和 $\boldsymbol{V}$。多头注意力机制的输出结果可以表示为：

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V}, \tag{2.16}$$

其中 d_k 为矩阵 $\boldsymbol{K}$ 中序列每一项的维度，经过 softmax 归一化处理后得到的权重表示对应词与文本中其他每个单词之间的相关度分数，与矩阵 $\boldsymbol{V}$ 中每一项加权求和即可得到最终的输出值。

多头注意力机制就是通过多组权重矩阵，生成多个独立的注意力模块，单独计算后将结果拼接在一起，形成最终输出，这样的设计主要是一种集成的作用，提升模型的泛化能力。

Transformer 结构中的前馈神经网络层即为两个线性全连接层拼接，提取特征的同时保持整个网络结构的输入、输出维度的一致，残差连接与归一化（Add

& Norm）的作用是将该层的输入、输出结果相加后，进行层归一化处理，起到加速收敛的作用。

在 Transformer 结构中，会有 6 个 Transformer Encoder 模块拼接在一起，前一个模块的输出将作为后一个模块的输入，最终输出的结果即为提取好的特征。这个特征可以由 Transformer Decoder 模块译码，进行后续的目标文本生成任务，也可以用来连接分类模型，完成文本分类任务。

2.4 BERT 预训练模型

2018 年，Devlin 等人[17]提出了一种应用了 Transformer Encoder 结构的模型：Bidirectional Encoder Representation from Transformer（BERT）。这是一个利用双向 Transformer Encoder 提取文本特征的预训练模型。它可以看作是类似 Word2Vec 一样，在特定训练任务下更新模型权重后，将其隐层表示作为词特征向量的一种神经网络，可以用于提取文本特征。和 Word2Vec 相比，BERT 模型层次更深，更加复杂，其所完成的预训练任务也有一定的提升，通过利用 Transformer 中的 Attention 机制和双向网络结构，更是能提取出深层次的语义信息。BERT 一经推出，就在多个自然语言处理领域的任务中取得重要突破，被称为是自然语言处理领域的集大成者。

“预训练”（pre-training）机制是 BERT 一大特色。使用 BERT 模型，首先需要在大规模语料中完成预训练任务，更新网络参数，随后，根据不同的下游任务，在 BERT 后连接相应的模型，通过“微调”的方式，可以让 BERT 训练出来的特征快速地适应于特定的任务。在第三章中，我们将重点探讨当“文本分类”作为下游任务时 BERT 预训练的表现方式。下面将主要介绍 BERT 模型的结构以及预训练任务。

BERT 的内部结构是由 12 个 Transformer Encoder 双向连接组成，每个 Transformer Encoder 中又包含 12 个注意力头，基本的网络结构如图 2.6 所示。其中，[CLS] 和 [SEP] 标记分别标识句子的起始和结束。具体的作用将在第三章介绍。

文本在输入 BERT 模型前需要先转化为词向量，BERT 中的词向量由 3 部分组成，词向量（Word Embedding）、位置向量（Positional Embedding）和分割向量（Segment Embedding）。其中位置向量和 Transformer 模型中有所区别，是通

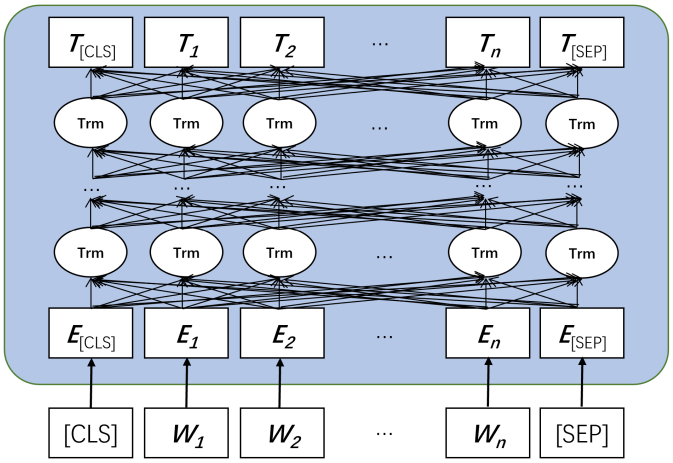

图 2.6 BERT 基本结构

过网络学习的方式得到，而非直接使用三角函数的线性组合进行表示。分割向量的作用和预训练有关，在训练 BERT 预训练任务中，通常需要两个句子拼接在一起作为输入，分割向量为了标识单词来自于第一句还是第二句。BERT 取三种向量之和作为模型输入。

BERT 模型会将输入依次通过 12 个 Transformer Encoder 结构，取最终输出为提取的特征。在预训练阶段，BERT 模型则会采用 2 中特定的预训练任务，在大规模语料中训练模型参数，分别为覆盖语言模型（Masked Language Model, MLM）和下一句预测（Next Sentence Prediction, NSP）。

MLM 任务是将输入句随机遮盖或替换部分词，通过模型根据上下文预测被遮盖或替换后的部分。实际操作中，会随机取句子中 15% 的单词进行处理，其中 10% 的单词被随即替换为其他词，10% 的单词不做处理，剩下 80% 的单词将被覆盖。这样做的目的是让模型能够通过上下文的语义关系进行纠错处理，更好地学习语义特征。

NSP 任务则是判断输入的两个句子是否为连续的上下句。训练时会将两个完整的句子拼接在一起输入模型，其中有的为连续的上下句，有的为随机选取的句子，模型需要判断句子的连续性。

将两种预训练任务结合后，我们就可以得到 BERT 模型预训练的与训练学习模式，即将两个句子拼接后，随机遮盖或替换掉部分词，然后输入至模型中，模型需要同时判断两个句子的连续性，以及预测被遮盖和替换的单词，并将两个任务的损失值相加，作为预训练的损失值，以此来训练 BERT。

BERT 模型提出了一种通用的解决自然语言处理问题的结构架式，Google 的官网也给出了一系列已经在大规模语料中预训练好的模型样本供我们使用，[1]使用起来较为方便。BERT 应用场景广泛，模型泛化能力强，拥有强大的特征提取能力，同时又十分灵活。在诸多自然语言处理任务中，BERT 模型已经达到了领先水平，而文本分类任务作为其中的一个应用，其性能也会得到显著的提升。

1 Google 官方对 BERT 的介绍和模型下载可以查看https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

第三章　BERT 应用于文本分类

BERT 模型为自然语言处理领域的任务提供了一种通用的处理方法，因此，在使用 BERT 模型处理文本分类任务时，仅需在 BERT 预处理模型后接上相关下游任务的模型，然后使用预训练好的参数，通过微调的方式，即可完成整个模型的训练。

Google 给出的基本 BERT 预训练模型参数是在通用的文本语料中训练而得的，其中包含 BooksCorpus 和维基百科共约 3.3G 的文本数据，模型参数 110M[17]。[1]在通用语料库中训练出的模型对大部分自然语言处理任务都有较好的适应性，也体现了 BERT 强大的泛用性。

使用 BERT 模型处理文本分类任务时，模型的一般结构如图 3.1 所示，待分类的文本将在分词后，通过 BERT 模型提取特征后，将 [CLS] 标记所对应的特征向量作为文本的特征表示，输入分类模型进行分类预测。

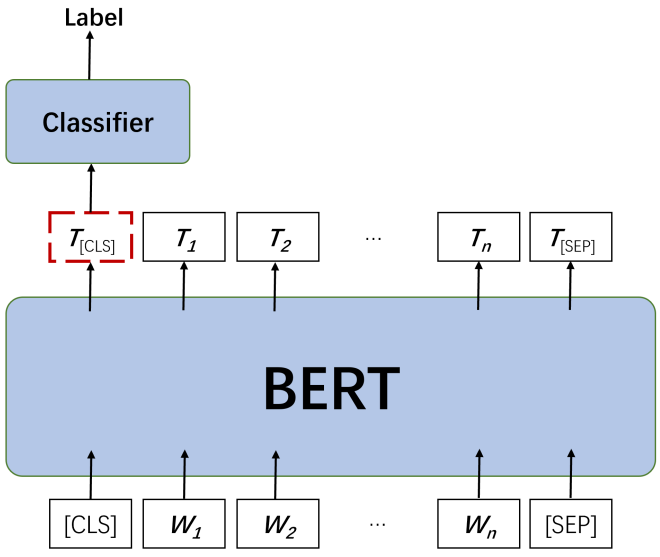

图 3.1　BERT 用于文本分类的基本结构

使用 [CLS] 标记作为文本特征的主要原因是：由于 BERT 内部是由多个包

1　BERT 开源仓库地址https://github.com/google-research/bert

含 Attention 机制的 Transformer Encoder 结构组成的，根据 Attention 的计算原理，任意一个单词在计算自注意力权重时，它和自己的注意力权重一定是最大的，因此如果使用具体的某一个单词的特征作为全文的特征，这个单词本身所占的权重就会增加，在模型训练时容易产生误导。

分类模型可以是合适的任意模型，它的最后一层一般情况下为从特征维度映射到类别数目维度的线性层。文本特征在经过分类模型输出后，需要经过 softmax 输出最终类别的概率分布，以数学公式的形式表示即为：

$$p(y|\boldsymbol{T}) = \text{softmax}(f(\boldsymbol{T}_{[\text{CLS}]})), \tag{3.1}$$

其中，f 为分类模型，softmax 是一种将权重数值的向量映射到离散概率分布的函数，单标签分类任务中，预测取概率分布中最大的值所对应的类别作为预测类别；训练过程中，可以将概率分布与实际类别的独热编码的交叉熵作为损失函数，对参数进行迭代。在论文的实验部分，我们将使用线性模型和 RNN 结构模型作为分类模型分别测试分类表现。

结合 Sun 等人[5]所给出的 BERT 模型在文本分类领域的使用方法，结合使用实际，我们给出 BERT 预训练模型应用于文本分类的一般方法：

（1）在大规模语料中对 BERT 模型进行预训练；

（2）对训练和测试集数据进行分词等文本预处理工作，然后进行索引、长文本截断等步骤；

（3）使用训练集数据，在基础通用 BERT 模型参数的基础上，进行 “增强预训练”，更新 BERT 模型参数；

（4）连接分类模型，基于增强预训练后的 BERT 模型参数，对模型进行 “微调”，进一步训练模型至给定训练步数阈值。

（5）训练完毕后，在测试集上对模型分类性能进行测试。

3.1 文本预处理

使用 BERT 进行文本分类时，对于文本的分词由 2 个部分组成：基础分词（Basic Tokenize）和单词片分词（Wordpiece Tokenize）两个阶段。

基础分词会对文本序列做一些基本的处理，主要包含 3 个阶段：首先将输

入转化为 unicode 字符串，并去除无法表示的字符（如空白符以外的控制字符），将所有空白符转化为空格；然后在中文字符前后加上空格，按照空格进行第一次分词；最后对音调字符进行调整（如 ē 修改为 e），同时将所有标点进行切分，并消除所有空白符。

单词片分词将对基础分词部分输出的单词列表进行进一步的切分，根据词典，按照从左到右，最长匹配的原则进行切分，消除不在词典中的未知单词。以这种方式分出的单词，除首部外，其他部分会在开头加上 “##” 进行标识。例如对单词 “bryant” 进行切分，按照最长匹配原则发现 “br” 在词典中，于是将单词切分为 “br” 和 “##yant”，对于 “##yant” 同理继续切分为 “##yan” 和 “##r”，因为 “##r” 在词典中，所以切分结束，最终单词被切分为三个部分 “br”，“##yan” 和 “##t”。单词片分词最关键的一个作用是还原单词的原始形态，例如对名词复数形态、动词的过去分词、现在分词形态等进行还原，同时保留原文本中这些形态的相关信息。分词结束后，将得到由一系列单词序列组成的列表，列表中的每一个单词被称为元素（token）。

使用 BERT 模型时，我们没有进行停用词的去除。“去除停用词” 广泛应用于基于统计概率方法的模型中，比如 TF-IDF 模型、贝叶斯分类器等，这样做可以去除大量对文本特征提取贡献不大的单词，但是在 BERT 中并没有必要。这是因为 BERT 模型内部的结构会自然学习对文本特征提取有用的关键词，因此无需再手动去除这些单词。同理，我们在进行中文分词时，也仅进行了非常基础的 “按字分词”，剩下的语义关系由 BERT 模型来进行提取。

3.2 长文本截断

由于 BERT 基础模型仅支持最长序列长度为 512 的输入，因此对于较长的文本，需要进行截断处理。Sun 等人[5]对截断方式进行了研究，指出了文本分类中最佳的截断方式为选择开头 128 个元素和末尾 382 个元素，并在开头和结尾分别加入 [CLS] 和 [SEP] 标签。

截断完成后，根据词典中的映射，我们将所有元素对应为唯一的数字编号，这个编号称为元素的索引（index），经过索引后，文本将由一个维度最大为 512 的向量表示。为了方便训练时对数据进行批处理，在预处理部分我们将统一各

个文本的维度为数据集中维度的最大值。对于维度不足最大值的样本，我们会在末尾添加 [PAD] 标签直到所有样本维度一致。[PAD] 标签所对应的索引值为 0，在 BERT 模型内部的 Attention 权值计算时会被屏蔽掉。

3.3 增强预训练

增强预训练（Further Pre-training）是一种提升模型分类性能的方法。基础的 BERT 模型是在通用的大规模语料中训练而得的，而短文本分类所使用的数据与通用语料在语言表述、文章结构等方面都有所不同，也就是说，这两类数据集拥有不同的数据分布，因此可以通过在所用训练集数据中“增强预训练”的方式，使得模型的数据分布更加接近我们所使用的数据。

增强预训练的方法就是应用和文本分类任务相关的数据集，执行 BERT 预训练所采用的两种无监督预训练任务，即 MLM 和下一句预测。Sun 等人[5]给出了三种增强预训练的方法并给出了比较：一是任务内（within-task）增强预训练，即使用分类任务的训练集本身对 BERT 进行增强预训练，这种方式训练所使用的数据集较为直接，并且在 100k 步后就可以达到最佳表现，在三种预训练方式中效果最好；二是领域内（in-domain）增强预训练，论文将常见的文本分类任务数据集分成三个领域：话题类、情感类和问答类，领域内增强预训练就是使用同一领域的数据集，对 BERT 模型进行进一步预训练；三就是领域间（cross-domain）增强预训练，即使用不同领域的数据集进行进一步预训练，论文中的实验表明，这种预训练方式并不是十分有效。

在论文中，我们采用任务内增强预训练方法来对 BERT 模型进一步训练。根据任务所用数据集的差异，增强预训练的效果可能产生变化，为了尽可能提升增强预训练的效果，我们给出一种改进后的文本分类任务的语料的增强预训练方法：

（1）对于训练集数据中多数只包含一句内容的短文本数据（如 AGNews），采用 MLM 单任务预训练方法，否则采用 MLM+NSP 结合的预训练方法；

（2）对于非短文本数据，每两个句子为单位进行分割，组成连续句子语料，句子之间以 [SEP] 标记分隔。剩下处于句尾的单个句子之间随机组合，组成非连续句子语料；

（3）对连续和非连续句子样本进行平衡化处理，即随机拆分连续句子语料中的样本，组成非连续句子语料，直到两种语料样本数量数量相差不超过 1；

（4）使用第 3.1 节文本预处理方法索引样本，进行增强预训练；

（5）增强预训练 100k 步后，保存模型参数，供后续微调模型训练使用。

以上给出的增强预训练方法，其核心思想就是解决下一句预测任务的样本类别不平衡的问题，并尽可能地利用训练集中的所有数据。样本类别不平衡即类别标签数量差别过大，会影响数据的分布已经模型的拟合度。此处我们需要尽可能保证连续句子样本和非连续句子样本的数量尽可能一致，避免出现标签数量不均衡的问题。

3.4 模型微调

模型微调阶段，需要在 BERT 模型后接入对应的分类模型，并载入增强预训练阶段得到的 BERT 模型初始参数。让模型在训练集上以较小的学习率进行训练，训练至固定轮数（epoch）后结束。

BERT 基础模型的内部共设置了 12 层 Transformer Encoder 结构，每一层都将输出一个隐层表示，作为下一层的输入。Sun 等人[5]就各层的隐层表示进行了研究，图 3.2 展示了使用各层次作为文本特征时模型整体的分类错误率。

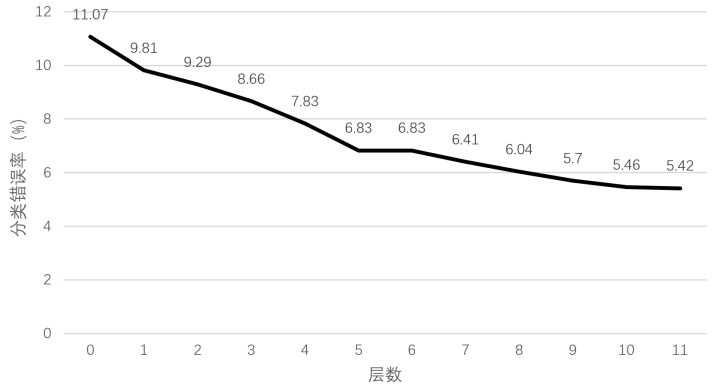

图 3.2 BERT 不同层对分类效果的影响

图中可以发现，较低层次的隐层表示所提取的特征，仅能学习文本的表层普遍特征，使用低层次隐层表示作为文本特征时，最终的分类表现将大幅降低。同时，文章也提到低层次的输出所包含的特征也有可能对整体文本特征的提取起到一定的作用，可以将 12 层隐层输出全部用于文本特征的表示。在第四章实

验部分，我们将尝试融合各层次的隐层表示作为文本特征，进行文本分类的任务，并进行比较。

第四章　实验过程与结果

本章将介绍论文的实验设置和实验结果，并对结果进行讨论。根据第三章的论述内容，我们将就 BERT 模型与文本分类中的两大问题进行实验验证，分别为：

问题一：利用 BERT 模型进行文本分类任务时，在仅使用 BERT 最后一层隐层输出，以及按一定比例混合全部 12 层隐层输出作为文本特征表示两种情况下，哪种情况模型的分类效果更好?

问题二：利用 BERT 模型进行文本分类任务时，执行第 3.3 节提出的“增强预训练”算法后再进行模型微调，是否比直接进行模型微调时的模型分类效果更好?

为了验证问题一，我们在一种文本数据集上进行验证。在模型的隐层表示部分，我们分别按照以下三种方式表示文本特征（1）直接取最后一层隐层输出；（2）取 12 层隐层输出平均；（3）利用网络学习 12 层隐层输出的权重，取加权平均。取分类效果最佳的方法，作为问题二 BERT 系列模型的文本特征表示方法。

为了验证问题二，我们在四种不同类型、不同语言的文本数据集上进行验证。我们比较相同参数设置下，使用增强预训练算法与不使用该算法时模型的分类效果。我们将设置两种分类模型进行对比，并且将此效果与其他论文中使用过的经典模型的效果进行对比，进一步说明 BERT 模型应用于文本分类模型的有效性。

4.1　实验准备

4.1.1　实验环境

本实验的模型训练测试任务在一台 GPU 服务器上运行，具体的硬件环境如表 4.1 所示。

本实验代码所使用的编程语言为 Python3.8，所使用的主要第三方包如表 4.2 所

表 4.1 实验硬件环境配置

环境	具体配置
CPU	7 Core Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz
RAM	30GB
GPU	RTX 3090 / Memory 24GB
操作系统	Ubuntu Linux version 5.4.0-90-generic

示。

表 4.2 实验编程环境配置

包	版本	功能描述
pytorch	1.10.0	搭建深度学习模型、执行模型训练与测试脚本
pytorch_pretrained_bert	0.6.2	加载 BERT 预处理方法和模型结构、参数等
numpy	1.21.5	向量、数组数据初步处理

4.1.2 数据集与分类模型

本实验所采用的 4 种文本数据集如表 4.3 所示。所选用的 4 种数据集均为经典的新闻、评论类短文本数据集，涵盖情感、话题两种不同的标签类别，以及中文、英文两种不同的语言。

表 4.3 实验所用数据集

数据集	语言	训练集数据量	测试集数据量	类别数	增强预训练方法
IMDb[5]	英文	25000	25000	2	MLM+NSP
Yelp[22]	英文	100000	10000	2	MLM+NSP
AGNews[23]	英文	120000	7600	4	MLM
SogouNews[5]	中文	54000	4000	6	MLM+NSP

为了验证 BERT 模型的分类性能，我们采用两种分类模型，分别进行模型训练和测试。

第一种模型采用 [CLS] 标签对应的特征输出作为文本特征表示，并将其输入至线性分类层，经过 softmax 后得到各类别的预测概率分布，得到最终的预测标签。我们称这种分类模型为 BERT-LINEAR 模型，该模型如图 4.1 所示。分类过程以数学公式的形式表示即为：

$$p(y|\boldsymbol{T}) = \text{softmax}(\boldsymbol{W}\boldsymbol{T}_{[\text{CLS}]}), \tag{4.1}$$

其中，$\boldsymbol{W}$ 为线性层参数，其将文本特征 $\boldsymbol{T}_{[\mathrm{CLS}]}$ 从特征维度映射至标签类别数的维度，得到最终的预测概率分布。

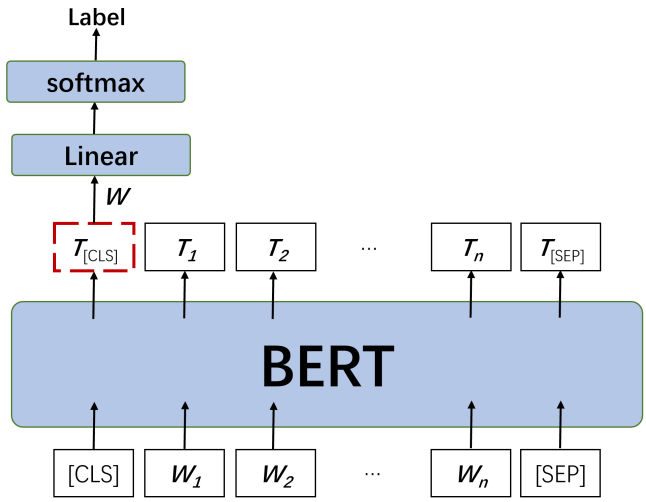

图 4.1 BERT-LINEAR 模型

第二种模型利用所有单词对应的特征输出作为文本特征，并将其看作一个序列，应用带有循环结构的双向 LSTM 分类模型，经过线性映射与 softmax 后得到各类别的预测概率分布与最终的预测标签。我们称这种接入 RNN 结构的分类模型为 BERT-RNN 模型，该模型如图 4.2 所示。分类过程以数学公式的形式表示为：

$$\boldsymbol{h}_1 = \overrightarrow{LSTM}(\boldsymbol{T}), \tag{4.2}$$

$$\boldsymbol{h}_2 = \overleftarrow{LSTM}(\boldsymbol{T}), \tag{4.3}$$

$$p(y|\boldsymbol{T}) = \mathrm{softmax}(\boldsymbol{W}[\boldsymbol{h}_1; \boldsymbol{h}_2]), \tag{4.4}$$

其中，$\boldsymbol{h}_1$ 和 $\boldsymbol{h}_2$ 分别为正向和反向 LSTM 模型的输出，将其拼接后经过线性层 $\boldsymbol{W}$ 映射至标签类别数维度，得到最终的概率分布。

在模型训练的过程中，将通过“增强预训练（FtP）”和“模型微调（FiT）”两个阶段。在使用不同分类模型、应用不同数据集的情况下，两个训练阶段所使用的参数完全一致，Sun 等人的研究中[5]实际表现最好的参数作为本实验的参数

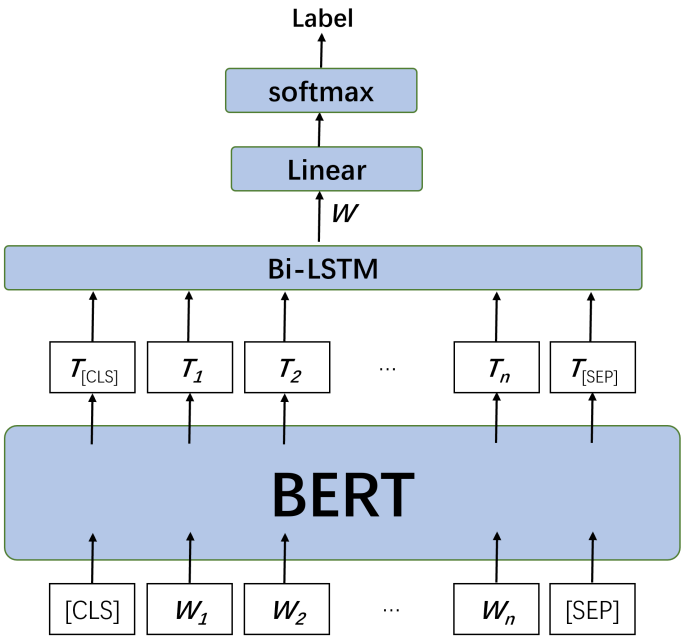

图 4.2 BERT-RNN 模型

设置。表 4.4 展示了两个训练阶段所使用的参数信息。

表 4.4 模型训练参数设置表

训练阶段	增强预训练 (FtP)	模型微调 (FiT)
损失函数	CrossEntropy	CrossEntropy
优化方法	Adam	Adam
学习率	5e-5(warmup: 0.1)	2e-5(warmup: 0.1)
迭代终止条件	10,0000 steps	4 epochs

4.2 实验结果

4.2.1 BERT 各层输出不同结合方式的分类结果

对于问题一，我们在 IMDb 数据集上进行验证。在使用 BERT-LINEAR 模型，并先后进行 FtP 和 FiT 两个训练阶段的情况下，比较选用 BERT 预训练模型的不同层隐层表示作为文本特征表示的情况下，模型的分类准确率。三种隐层表示的选取方式分别为：

方法一：选用第 12 层（最后一层）的隐层表示 $\boldsymbol{T}_{12}$ 作为文本特征表示；

方法二：取 1 至 12 层隐层表示的平均值作为文本特征表示；

方法三：使用可训练参数 $\alpha_1, ..., \alpha_{12}$，初始值设置为 $\frac{1}{12}$，取 1 至 12 层隐层表

示的加权和 $\sum_{i=1}^{12} \alpha_i \boldsymbol{T}_i$ 作为文本特征表示，训练参数 α_i 将会随着训练过程更新参数。

模型分类结果将用宏-F1 分数的指标进行衡量。表 4.5 展示了三种情况下模型的分类 F1 分数。

表 4.5 IMDb 数据集下使用不同文本特征表示的 BERT-LINEAR 模型分类 F1 分数表

特征表示	F1 分数
（1）只选用最后一层	**0.945**
（2）选用 1 至 12 层的平均值	0.943
（3）设置可学习权重参数	0.942

从表 4.5 的结果来看，三种方法对应的分类效果差别不大，仅使用最后一层时模型的分类效果有略微的提升。基于此实验结果，在验证问题二时，我们统一采取“仅使用 BERT 最后一层隐层输出作为文本特征表示”进行模型搭建。

4.2.2 BERT 经过增强预训练阶段的分类结果

对于问题二，我们在表 4.3 所展示的 4 中数据集上进行验证。我们分别使用 BERT-LINEAR 和 BERT-RNN 模型，并分别在使用默认预训练参数和增强预训练后的参数两种情况下，比较模型的分类结果。模型结果将分别用宏-F1 分数和分类错误率作为评价指标，在使用分类错误率时，我们还将与其他文献中的经典模型的分类错误率进行对比，验证 BERT 模型的分类优势。

表 4.6 和表 4.7 分别展示了不同模型使用不同预训练方式时模型的宏-F1 分数和分类错误率。[1]

表 4.6 文本分类数据集使用 BERT 预训练模型分类的宏 F1 分数

模型	IMDb	Yelp	AGNews	SogouNews
BERT-LINEAR+FiT	0.945	0.944	0.933	0.957
BERT-RNN+FiT	0.944	0.944	0.911	0.944
BERT-LINEAR+FtP+FiT	0.945	**0.946**	**0.940**	**0.962**
BERT-RNN+FtP+FiT	**0.946**	**0.946**	0.915	0.949

从表中可以看出，经过增强预训练后的 BERT 模型分类性能有略微提升，使用线性分类层和 RNN 分类层时，根据数据集的不同有一定差异，但整体而言，BERT 预训练模型的结果优于 TextCNN、TextRNN 等一般模型的结果。

1 有 * 标记注释的结果为文献中存在该数据集的结果，但由于所使用数据集等差异（如标签的生成方式）而改为自己实现的结果，符号“/”表示文献中没有使用此数据集的结果

表 4.7 文本分类数据集使用经典模型与 BERT 的分类错误率（%）

模型	IMDb	Yelp	AGNews	SogouNews
Char-level CNN[24]	/	11.87*	9.51	4.70*
Standard LSTM[25]	8.90	7.98*	6.95*	/
Skim-LSTM[25]	8.80	/	6.40	/
D-LSTM[26]	/	7.40	7.90	5.10
BERT-LINEAR+FiT	5.51	5.62	6.66	4.28
BERT-RNN+FiT	5.64	5.59	8.92	5.60
BERT-LINEAR+FtP+FiT	5.46	**5.37**	**6.04**	**3.78**
BERT-RNN+FtP+FiT	**5.37**	**5.37**	8.46	5.13

4.3 结果讨论

4.3.1 BERT 不同层特征对分类结果的影响

在对问题一的验证中，我们发现仅选用最后一层的隐层表示 $\boldsymbol{T}_{12}$ 作为文本特征时，分类结果有略微优势。实际上，IMDb 数据集的分类内容为评论情感倾向，这要求模型学习文本中的情感信息。这是模型深层输出所具有的特征。

Ganesh Jawahar 等人[27]探究 BERT 预训练模型各层所学习到的内容。论文使用“探针任务”，在 BERT 的不同层接入分类器过使用不同层的输出作为文本特征完成这些任务。这些探针任务被分类成了“表层任务”、“语法任务”和“语义任务”。他们发现，浅层的表示在完成“表层任务”和“语法任务”更加有优势，而深层的表示更善于完成“语义任务”。论文还将一系列拥有不同语法结构的语法短语输入到 BERT，通过对这些短语的各层表示进行聚类分析，发现浅层表示能够清晰地区分不同语法结构的短语，而深层次的表示不具备这样的能力。

对于网络上流行的新闻、评论类短文本，在使用 BERT 模型进行分类时，加入语法等浅层特征表示意义不大，并且有可能产生一定的误导。因此，只使用深层网络结果作为文本特征的想法是合理的。

在之前的实验中，我们尝试通过添加网络参数，让模型自主学习各层表示所占的权重。权重的初始值设置为相同权重，表 4.8 展示了微调结束后，各层特征表示所占权重的结果。从中可以明显的看出，在“微调”阶段学习率较低的情况下，权重依然较为明显地向深层倾斜，这也证明了深层的特征表示更适合处理语义相关的分类任务。

表 4.8 BERT 各层特征所占权重数值

层数	1	2	3	4	5	6
权重	0.083	0.085	0.087	0.089	0.090	0.090
层数	7	8	9	10	11	12
权重	0.090	0.090	0.095	0.107	0.104	0.100

4.3.2 BERT 在低样本量数据集情况下的分类表现

从问题二的实验结果可以看到，BERT 预训练模型对于样本量较少的 IMDb 数据集，其分类性能较基础模型有明显的提升。BERT 预训练模型拥有从数据集快速收敛的能力，往往在第一个 epoch 训练期间，损失函数值已经趋近于局部最优点。图 4.3 展现了使用 TextRNN 结构和 BERTLRN 模型在 IMDb 数据集训练过程中，损失函数值在第 1 个 epoch 的变化趋势，图中每一个批（batch）的数据量为 16。

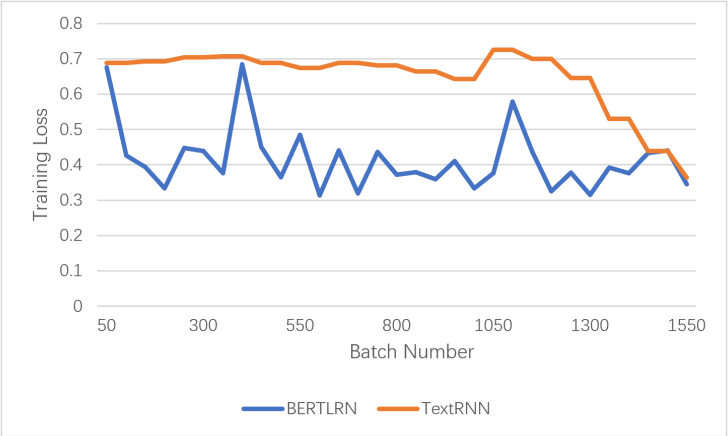

图 4.3 BERT-LINEAR 和 TextRNN 在第一个 epoch 的训练损失值

从图中可以看出，BERT 模型在微调很早阶段，损失就已经进入振荡下降趋势，而一般的 TextRNN 结构模型需要一定的“启动时间”，图中 TextRNN 模型经过了约 1000 个 batch（即约 16000 条数据）后，才呈现下降趋势。这一趋势的呈现与 BERT 模型的“预训练”这一特点相关。

在一般的模型中，影响收敛速度的主要因素包括模型的参数初始化值。不同的初始化参数会对模型训练过程的损失值的变化产生的一定影响，甚至影响模型最后落入不同的“局部最优点”。和其他模型不同，BERT 模型需要经过大规模预料的预训练和特定语料上的“微调”才能使用。“预训练”这个步骤就可

以看作是通过训练的方式寻找模型的“最佳初始化参数”。BERT 依靠以大规模文本数据和算力为代价，寻找到了一个适合几乎所有自然语言处理领域任务的初始化参数，任何文本挖掘任务仅需要以很小的学习率调整 BERT 模型参数，就可以使模型损失快速收敛到局部最优点，同时，相比较其他模型选取初始模型参数的方法，BERT 模型的与训练参数还保留了对文本语法、语义等深层信息的学习能力。这也是 BERT 在自然语言处理领域如此强势的一个重要原因。

BERT 收敛速度之快的一个优点就是 BERT 在低样本量数据集的情况下分类表现优于其他模型。在训练集数据较少的情况下，BERT 模型依靠预训练参数的学习能力，依然能够快速地收敛模型，同时保证模型的泛化能力，不至于欠拟合。实验中 IMDb 数据集上的结果也说明了这个观点。

虽然 BERT 在低样本量数据集下表现较好，但这并不表明 BERT 在微调阶段就不需要大量样本数据了。任何深度学习模型都是以大数据作为支撑的，BERT 也不例外，更何况 BERT 已经在大规模语料中完成了预训练的任务。但是在样本量不大的前提下，我们可以利用“增强预训练”最大程度的提高 BERT 在低样本数据集的分类表现。

“增强预训练”的一个目的是将 BERT 模型参数从标准文本数据集分布向目标数据集分布偏移的过程。从特征提取的角度，这可以使 BERT 模型所提取的文本特征更加适合于目标任务；从模型参数的角度，它也为微调训练阶段提供了一个更适合目标任务的初始模型参数。

Sun 等人[5]也提到了 BERT 模型在低样本数据集下的分类表现。他们指出，BERT 模型应用于低样本数据集使，其分类性能能够得到显著提升，应用了“增强预训练”后，这种提升幅度还会进一步扩大。它们甚至在低样本数据集中再次缩减样本量，比较不同样本量数据情况下模型的分类结果。虽然无论是否进行过增强预训练，分类错误率均随样本量的减小而增大，但增强预训练后的模型错误率普遍高于使用基本参数的情况。特别是在只取 0.4%（100 条）样本时，增强预训练后的模型分类错误率比使用标准参数模型少了约 8%。因此，我们不难发现，BERT 预训练模型在低样本量数据中有较好的分类表现，“增强预训练”可以进一步提高模型的分类性能。

第五章　总结

5.1　本文主要内容

本文介绍了短文本分类的基本原理和主要方法，通过结合 BERT 预训练模型，探讨使用 BERT 模型进行文本分类的基本方法，并对 BERT 文本分类模型的分类性能进行验证。论文基于 BERT 文本分类的一般性方法，通过在四个数据集上对 BERT-LINEAR 和 BERT-RNN 模型进行训练、测试，比较验证了 BERT 预训练模型在文本分类领域的优势，并对 BERT 模型的特点进行了进一步的分析。通过一系列实验验证和分析讨论，论文主要得出了以下结论：

首先，BERT 预训练模型得益于它在大规模文本语料中的预训练特点，使用无监督学习方式，提供了一种优于传统模型的，能够学习文本深层语义信息的端到端分类方式，使用方便、灵活，拥有较强的泛化能力。基于其“文本预处理-增强预训练-模型微调”的策略，我们对于任务内增强预训练方法，我们改进了基本算法，用于解决语料样本平衡度和不同长度文本的适应性问题。

其次，BERT 的不同层特征代表了学习文本不同特征的能力，其中浅层的隐层输出体现了对文本表层知识和语法结构的学习，深层的隐层输出则体现了对文本深层语义知识的学习，文本分类任务需要基于文本的语义信息进行分类判断，因此需要使用较深层模型输出作为文本特征进行分类。

最后，BERT 模型在微调训练过程中能够以较快的速度进入震荡收敛阶段，收敛速度快，对低样本量的文本数据有着较好的分类效果，同时“增强预训练”的过程也证实能够进一步提升模型的低样本量数据的分类效果，并进一步提升模型的分类性能。

5.2　不足之处与展望

本文也存在一些不足之处，需要进一步完善。本文的实验结果中，BERT 在应用线性模型和 RNN 结构模型进行分类时，在不同的数据集上两种模型各有优

劣，而非某种结构全部优于另一种。本文没有对 BERT 模型适合于文本分类的结构做出更深的研究。同时，本文在预训练阶段提出了提升模型分类性能的方法，但是在微调阶段并没有做出其他的改进，后续可以在模型微调阶段寻找更加合适的训练方法。

BERT 模型自 2018 年诞生以来，频频在多个自然语言处理领域取得成就，覆盖了多个文本挖掘任务，BERT 也因此被称为自然语言领域 “具有开创性的自然语言处理架构”。BERT 的出现得益于硬件算力的提升和大数据的发展，并且为后续端到端的自然语言处理方法奠定了基础。在后续的研究中，我们可以深入探究 BERT 的工作原理，提出适合于各种自然语言处理任务的通用的、高效的处理方法。

参考文献

[1] PATGIRI R, AHMED A. Big Data: The V's of the Game Changer Paradigm [C]//2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS). 2016: 17-24.

[2] HIRJI K K. Discovering Data Mining: From Concept to Implementation[J]. SIGKDD Explor. Newsl., 1999, 1(1): 44-45.

[3] 张宇豪. 基于 BERT 的新闻短文本分类方法研究[D]. 2021.

[4] Aggarwal, CharuC. Data Mining: The Textbook[M]. Data Mining: The Textbook, 2015.

[5] SUN C, QIU X, XU Y, et al. How to Fine-Tune BERT for Text Classification?[C]//SUN M, HUANG X, JI H, et al. Chinese Computational Linguistics. Cham: Springer International Publishing, 2019: 194-206.

[6] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient Estimation of Word Representations in Vector Space[C]//BENGIO Y, LECUN Y. 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings. 2013.

[7] SHAH F P, PATEL V, Ieee. A Review on Feature Selection and Feature Extraction for Text Classification[C]//IEEE International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). 2016: 2264-2268.

[8] JOLLIFFE I T, CADIMA J. Principal component analysis: a review and recent developments[J]. Philosophical Transactions of the Royal Society a-Mathematical Physical and Engineering Sciences, 2016, 374(2065).

[9] NIHARIKA S, LATHA V S, LAVANYA D R. A Survey On Text Categorization [J]. International Journal of Computer Trends & Technology, 2012, 3(1).

[10] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in neural information processing systems, 2013, 26.

[11] 张欣. 基于多尺度 CNN 与 LSTM 混合模型的中文新闻分类研究[D]. 2021.

[12] ZOLNIEREK A, RUBACHA B. The Empirical Study of the Naive Bayes Classifier in the Case of Markov Chain Recognition Task[C]//KURZYŃSKI M, PUCHAŁA E, WOŹNIAK M, et al. Computer Recognition Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005: 329-336.

[13] WANG L. Support Vector Machines: Theory and Applications[M]. Machine Learning, 2001.

[14] KALCHBRENNER N, GREFENSTETTE E, BLUNSOM P. A Convolutional Neural Network for Modelling Sentences[C]//52nd Annual Meeting of the Association-for-Computational-Linguistics (ACL). 2014: 655-665.

[15] LIU P, QIU X, HUANG X. Recurrent Neural Network for Text Classification with Multi-Task Learning[J]. AAAI Press, 2016.

[16] HOCHREITER S, SCHMIDHUBER J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[17] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 4171-4186.

[18] VASWANI A, SHAZEER N, PARMAR N, et al. Attention Is All You Need[C]// Advances in Neural Information Processing Systems: 31st Annual Conference on Neural Information Processing Systems (NIPS): vol. 30. 2017.

[19] YANG X, YANG L, BI R, et al. A Comprehensive Verification of Transformer in Text Classification[C]//Lecture Notes in Artificial Intelligence: 18th China National Conference on Computational Linguistics (CCL): vol. 11856. 2019: 207-218.

[20] 周志华. 机器学习[M]. 清华大学出版社, 2016.

[21] ZHANG Y, JIN R, ZHOU Z H. Understanding bag-of-words model: a statistical framework[J]. International Journal of Machine Learning and Cybernetics, 2010, 1(1-4): 43-52.

[22] RAYANA S, AKOGLU L. Collective Opinion Spam Detection: Bridging Review Networks and Metadata[C]//KDD '15: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Sydney, NSW, Australia: Association for Computing Machinery, 2015: 985-994.

[23] ANAND A. AG News Classification Dataset[Z]. Online Database. 2020.

[24] ZHANG X, ZHAO J, LECUN Y. Character-Level Convolutional Networks for Text Classification[C]//NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. Montreal, Canada: MIT Press, 2015: 649-657.

[25] SEO M, MIN S, FARHADI A, et al. Neural Speed Reading via Skim-RNN [EB/OL]. arXiv. 2017. https://arxiv.org/abs/1711.02085.

[26] YOGATAMA D, DYER C, LING W, et al. Generative and Discriminative Text Classification with Recurrent Neural Networks[EB/OL]. arXiv. 2017. https://arxiv.org/abs/1703.01898.

[27] JAWAHAR G, SAGOT B, SEDDAH D. What Does BERT Learn about the Structure of Language?[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 3651-3657.

致　　谢

论文从去年 12 月开题到现在，已经过去了将近半年的时间，这半年来，无论是文献搜集、阅读，还是学习深度学习网络的训练方式，一直到最后分析、写作，离不开各位导师，同学，以及亲朋好友的帮助。

首先，两位导师对于论文的完成提供了非常重要的帮助。张建兵老师作为我进入 NLP 领域的引路人，从 2020 年 NJUNLP 夏令营开始，就一直在 NLP 领域给予了我很大的帮助，这次论文的撰写，在选题方面给予了我很多建设性的意见，奠定了论文的基础。宗石老师在我实验和论文撰写阶段，从学术严谨的角度提出了很多非常有用的建议，指正了我实验过程中的许多不当之处，使我得以提升论文内容的严谨性和丰富性。

其次，父母的支持，以及同学的帮助，也是这篇论文完成必不可少的因素。父母的一次次鼓励让我得以在我梦想的道路上坚定决心，正是他们的关心和日夜操劳，才能让我在这大学四年尽量不留遗憾。校园里朝夕相处的同学们，亦是我求学道路上最好的伙伴，他们的陪伴让我在这大学四年里不会孤单。

最后，感谢南京大学和计算机科学与技术系为我提供了一个求知的平台，让我在本科期间充分地发掘自己，朝着梦想的道路不断拼搏。南大不仅为我论文写作提供了一系列的技术支持，也让我能够有机会与这么多优秀的老师、同学相识，接触先进的专业知识，实现自己的梦想。

在我执笔撰写论文之时，南京大学即将迎来一百二十周年校庆，特此祝贺南大百廿芳华永继，再创辉煌。

四年的本科时光即将结束，新的旅途即将开始。我将在求知的道路上继续前行，书写人生的全新篇章。